

CLAIMS

[What is claimed is:] We claim

1 1. (Amended) [A method of operating a computer system,
2 said computer system including at least one processor,
3 comprising.
4 establishing a plurality of memory units each
5 having a corresponding memory location;
6 executing a plurality of tasks running on said at
7 least one processor, said plurality of tasks being operable
8 to share data;
9 defining a plurality of lists for each memory
10 location;
11 determining the validity of said data in said
12 memory unit;
13 locking at least one of said plurality of lists
14 if said data is invalid;
15 inserting an entry corresponding to one of said
16 plurality of tasks onto said locked list;
17 unlocking said locked list; and
18 determining if data is inputted in said memory
19 location between said determining step and said unlocking
20 step.]

21 A processing system for performing addition and
22 subtraction within limits upon a shared value comprising:
23 means for performing a first uninterruptible
24 operation upon the shared value stored in an affected
25 reservation location, the first uninterruptible operation
26 using an operand;
27 means for comparing a resulting value of the
28 first uninterruptible operation stored in the affected
29 reservation location to limit values stored in limit
30 locations;
31 means for performing a second uninterruptible
32 operation to restore the affected reservation location if
33 the resulting value of the first uninterruptible operation
34 is not within the limit values in the limit locations;
35 means for reporting a failure if the resulting
36 value of the first uninterruptible operation is not within
37 the limit values in the limit locations;
38 means for performing a third uninterruptible
39 operation to update an actual value location if the
40 resulting value of the first uninterruptible operation is
41 within the limit values in the limit locations;
42 means for performing a fourth uninterruptible
43 operation to update an unaffected reservation register if

44 the resulting value of the first uninterruptible operation
45 is within the limit values in the limit locations; and
46 means for reporting a success if the resulting
47 value of the first uninterruptible operation is within the
48 limit values in the limit locations.

A¹
1 2. (Amended) [A method for operating a computer system,
2 said computer system comprising at least one process,
3 comprising:
4 establishing a plurality of memory units each
5 having a corresponding memory location;
6 running a plurality of tasks on said processor,
7 said plurality of tasks being operable to share data;
8 defining a plurality of lists for each memory
9 location;
10 inserting an entry corresponding to one of said
11 plurality of tasks onto one of said plurality of lists if
12 said one list is unlocked; and
13 determining if another of said lists is unlocked
14 if said one list is locked.]

15 The processing system of claim 1 wherein the first,
16 second, third, and fourth uninterruptible operations are
17 LOCK XADD operations.

1 3. (Amended) [A method for synchronizing processes in a
2 computer system, said computer system including at least
3 one processor, comprising:
4 establishing a plurality of memory units each
5 having a corresponding memory location
6 executing a plurality of tasks running on said
7 processor, said plurality of tasks being operable to share
8 data located in said memory units;
9 defining a plurality of lists for each memory
10 location;
11 locking at least one of said plurality of lists
12 if said data is not valid;
13 inserting an entry corresponding to one of said
14 plurality of tasks onto said locked list;
15 unlocking said locked list;
16 suspending said entered task until valid data is
17 found in said memory unit;
18 reading said valid data;
19 determining if other data is inputted in said
20 memory unit before said locking step and after said
21 unlocking step; and
22 reading said other data if it appears in said
23 memory unit.]

24 A processing system for performing addition and
25 subtraction within limits upon a shared value comprising:
26 means for receiving an operand;
27 means for performing a first uninterruptible
28 operation upon the shared value stored in an affected
29 reservation location, the first uninterruptible operation
30 using the operand;
31 means for comparing a resulting value of the
32 first uninterruptible operation stored in the affected
33 reservation location to limit values stored in limit
34 locations;
35 means for performing a second uninterruptible
36 operation to restore the affected reservation location if
37 the resulting value of the first uninterruptible operation
38 is not within the limit values in the limit locations;
39 means for and reporting a failure if the
40 resulting value of the first uninterruptible operation is
41 not within the limit values in the limit locations;
42 means for performing a third uninterruptible
43 operation to update an actual value location if the
44 resulting value of the first uninterruptible operation is
45 within the limit values in the limit locations;
46 means for performing a fourth uninterruptible
47 operation to update an unaffected reservation register if

48 the resulting value of the first uninterruptible operation
49 is within the limit values in the limit locations; and
50 means for reporting a success if the resulting
51 value of the first uninterruptible operation is within the
52 limit values in the limit locations.

1 4. (Amended) [The method of claim 3, wherein the locking
2 step further comprises activating selected other ones of
3 said plurality of tasks that are entered on said locked
4 list.]

A 5 The processing system of claim 3 wherein the first,
6 second, third, and fourth uninterruptible operations are
7 LOCK XADD operations.

1 5. (Amended) [The method of claim 3, wherein said
2 plurality of lists forms a linked list.]

3 A method for performing addition and subtraction
4 within limits upon a shared value comprising the steps of:
5 first, performing a first uninterruptible
6 operation upon the shared value stored in an affected
7 reservation location, the first uninterruptible operation
8 using an operand;
9 second, comparing a resulting value of the first
10 uninterruptible operation stored in the affected

11 reservation location to limit values stored in limit
12 locations;
13 third, performing a second uninterruptible
14 operation to restore the affected reservation location;
15 fourth, reporting a failure if the resulting
16 value is not within the limit values in the limit
17 locations;
18 fifth, performing a third uninterruptible
19 operation to update an actual value location if the
20 resulting value is within the limit values in the limit
21 locations;
22 sixth, performing a fourth uninterruptible
23 operation to update an unaffected reservation register if
24 the resulting value is within the limit values in the limit
25 locations; and
26 seventh, reporting a success if the resulting
27 value is within the limit values in the limit locations.

1 6. (Amended) [The method of claim 3, wherein said
2 plurality of lists is between four and eight.]

3 The method of claim 5 wherein the first, second,
4 third, and fourth uninterruptible operations are LOCK XADD
5 operations.

1 7. (Amended) [The method of claim 3, further comprising
2 transferring the operation of said locked list when said
3 locked list is locked by another one of said plurality of
4 tasks.]

5 A computer readable medium containing computer
6 readable code comprising:

7 a code segment for performing a first
8 uninterruptible operation upon the shared value stored in
9 an affected reservation location, the first uninterruptible
10 operation using an operand;

11 a code segment for comparing a resulting value of
12 the first uninterruptible operation stored in the affected
13 reservation location to limit values stored in limit
14 locations;

15 a code segment for performing a second
16 uninterruptible operation to restore the affected
17 reservation location;

18 a code segment for reporting a failure if the
19 resulting value is not within the limit values in the limit
20 locations;

21 a code segment for performing a third
22 uninterruptible operation to update an actual value
23 location if the resulting value is within the limit values
24 in the limit locations;

25 a code segment for performing a fourth
26 uninterruptible operation to update an unaffected
27 reservation register if the resulting value is within the
28 limit values in the limit locations; and
29 a code segment for reporting a success if the
30 resulting value is within the limit values in the limit
31 locations.

1 8. (Amended) [A computer system having enhanced
2 concurrency, comprising:

3 a plurality of processors;

4 a plurality of tasks running on said plurality of
5 processors;

6 a plurality of memory units each having a
7 corresponding memory location;

8 a plurality of lists corresponding to each of
9 said memory location;

10 wherein one of said plurality of tasks is
11 responsible for activating selected ones of said plurality
12 of tasks contained on the same list as said one task.]

13 A processing system for performing addition and
14 subtraction within limits upon a shared value comprising:

15 a processor, the processor

16 performing a first uninterruptible operation
17 upon the shared value stored in an affected reservation
18 location, the first uninterruptible operation using an
19 operand;
20 comparing a resulting value of the first
21 uninterruptible operation stored in the affected
22 reservation location to limit values stored in limit
23 locations;
24 performing a second uninterruptible
25 operation to restore the affected reservation location if
26 the resulting value of the first uninterruptible operation
27 is not within the limit values in the limit locations;
28 reporting a failure if the resulting value
29 of the first uninterruptible operation is not within the
30 limit values in the limit locations;
31 performing a third uninterruptible operation
32 to update an actual value location if the resulting value
33 of the first uninterruptible operation is within the limit
34 values in the limit locations;
35 performing a fourth uninterruptible
36 operation to update an unaffected reservation register if
37 the resulting value of the first uninterruptible operation
38 is within the limit values in the limit locations; and

39

reporting a success if the resulting value

40 of the first uninterruptible operation is within the limit

41 values in the limit locations.
